COURSE DESCRIPTION CARD - SYLLABUS

Course name
Basics of programming - Object-oriented languages [N1MiBM2>JOPP]

Course

Field of study
Mechanical Engineering

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
part-time

Requirements
elective

Number of hours

Lecture
8

Laboratory classes
16

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators	Lecturers

Prerequisites

Basic computer skills

Course objective

The aim of the course is to familiarize students with the basics of programming in high-level languages (using a selected language). During the classes, basic concepts and structures occurring in programming are presented (variables, data types, instructions, handling communication with the user, subroutines) as well as the construction of programs using the above structures (combined with the analysis of problems in an algorithmic context) and checking their correctness

Course-related learning outcomes

Knowledge:
Knows the typology of programming languages and has knowledge of their basic applications. Knows tools supporting a programmer's work. Knows sources of information covering the problems of creating, programming and using applications. Is able to formulate algorithms and program them using at least one of the popular tools.

Skills:
Can create simple applications. Can design a good user interface for applications. Is able to assess, at a

basic level, the usefulness of routine IT methods and tools and to select and apply the appropriate method and tools for typical IT tasks

Social competences:
Understands the need for lifelong learning; can inspire and organize the learning process of other people. Is aware of the importance and understanding of non-technical aspects and effects of engineering activities.

Methods for verifying learning outcomes and assessment criteria
Learning outcomes presented above are verified as follows:

Lecture: Assessment based on a written test consisting of 30 test questions on a scale of 0/1. Passing if a minimum of 51% is obtained.
Laboratory: Assessment is based on a practical test in laboratories worth 60 points and tasks carried out in the laboratory - 40 points. The final grade is determined according to the following scoring system: 100 points in accordance with the adopted grading system - 3.0 from 41 points, 3.5 from 56 points, 4.0 from 71 points, 4.5 from 81, 5.0 from 91 points

Programme content
Lecture: The lecture on "Scripting Languages" will present one programming language chosen by the instructor (either C or Python). The lecture is organized to be accessible for individuals with very basic programming skills (especially in the initial phase of the lecture), as well as useful for advanced individuals. The first part will cover the semantics and syntax of the language, basic instructions and operators, and a detailed explanation of loop operation principles. The later part will discuss methods of building simple and advanced data structures along with examples.
Laboratory - practical application of knowledge gained in the lecture, implementation and testing - topics covered:
1. Compilation model.
2. Operators and their precedence.
3. Preprocessor directives, macros.
4. Arrays.
5. Pointers and their relationship with arrays.
6. Passing values to functions.
7. Function pointers.
8. Structures.
9. Dynamic creation of data structures using pointers.
10. File operations.
11. Discussion of the standard C language library (functions related to basic string operations, mathematical functions, conversions).
12. Discussion of safe and unsafe functions (buffer overflow issues).
13. Writing large programs and the use of GNU make, autoconf, and configure.
14. Error search strategies and the use of debugger-type tools, valgrind.

Course topics

Teaching methods

Lecture: Multimedia presentation - illustrated with examples and videos, analysis of problems along with their visualization on the board, discussion, and problem analysis.
Laboratory: The laboratory will be conducted through computer work, during which programs illustrating the discussed topics will be written together with the instructor. In later stages of the laboratory sessions, students will individually write programs to accomplish specific tasks that have been previously discussed and demonstrated by the instructor.

Bibliography
Basic:
1. C. Albing, J.P. Vossen, C. Newham : Bash. Receptury. Helion, Gliwice.

2. H. Schwichtenberg : Windows PowerShell. Podstawy. Helion, Gliwice.
3. P. Norton, A. Samuel : Python. Od podstaw. Helion, Gliwice.
4. M. Lutz : Python. Wprowadzenie. Helion, Gliwice.

Additional:
Źródła internetowe: np. https://docs.python.org/3/library/index.html

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	24	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	76	3,00